第 39 卷第 10 期
录用定稿
计算机应用研究
Application Research of Computers
ChinaXiv合作期刊
Vol. 39 No. 10
Accepted Paper

基于 End-to-end 深度强化学习的多车场车辆路径优化 *

雷　坤[1]，郭　鹏[1,3]，王祺欣[1]，赵文超[1]，唐连生[2†]

(1. 西南交通大学 机械工程学院，成都 610031; 2. 宁波工程学院 经济与管理学院，浙江 宁波 315211; 3. 轨道交通运维技术与装备四川省重点实验室，成都 610031)

摘　要：为提高多车场车辆路径问题(Multi-Depot Vehicle Routing Problem，MDVRP)的求解效率，提出了端到端的深度强化学习框架。首先，将 MDVRP 建模为马尔可夫决策过程(Markov Decision Process，MDP)，包括对其状态、动作、收益的定义。同时，提出了改进图注意力网络(Graph Attention Network，GAT)作为编码器对 MDVRP 的图表示进行特征嵌入编码，设计了基于 Transformer 的解码器。并采用改进 REINFORCE 算法来训练该模型。该模型不受图的大小约束，即其一旦完成训练，就可用于求解任意车场和客户数量的算例问题。最后，通过随机生成的算例和公开的标准算例验证了所提出框架的可行性和有效性。即使在求解客户节点数为 100 的 MDVRP 上，经训练的模型平均仅需 2 毫秒即可得到与现有方法相比更具优势的解。

关键词：多车场车辆路径问题；深度强化学习；图神经网络；REINFORCE 算法；Transformer 模型

中图分类号：TP242.2　　doi: 10.19734/j.issn.1001-3695.2022.03.0095

End-to-end deep reinforcement learning framework for multi-depot vehicle routing problem

Lei Kun[1], Guo Peng[1,3], Wang Qixin[1], Zhao Wenchao[1], Tang Liansheng[2†]

(1. *School of Mechanical Engineering*, *Southwest Jiaotong University*, *Chengdu* 610031, *China*; 2. *School of Economics & Management*, *Ningbo University of Technology*, *Ningbo Zhejiang* 315211, *China*; 3. *Technology & Equipment of Rail Transit Operation & Maintenance Key Laboratory of Sichuan Province*, *Chengdu* 610031, *China*)

Abstract: This paper proposed an end-to-end deep reinforcement learning framework to improve the efficiency of solving the Multi-Depot Vehicle Routing Problem (MDVRP) . There is a novel formulation of the Markov Decision Process (MDP) for the MDVRP, including the definitions of its state, action, and reward. Then, this paper exploited an improved Graph Attention Network (GAT) as the encoder to perform feature embedding on the graph representation of MDVRP, and designed a Transformer-based decoder. Meanwhile, this paper used the improved REINFORCE algorithm to train the proposed encoder-decoder model. Furthermore, the designed encoder-decoder model is not bounded by the size of the graph. That is, once the framework is trained, it can be used to solve MDVRP instances with different scales. Finally, the results on randomly generated and published standard instances verify the feasibility and effectiveness of the proposed framework. Significantly, even on solving MDVRP with 100 customer nodes, the trained model takes only two milliseconds on average to obtain a very competitive solution compared with existing methods.

Key words: multi-depot vehicle routing problem; deep reinforcement learning; graph neural network; reinforce algorithm; transformer model

0　引言

随着电子商务和交通运输产业的不断壮大，物流业飞速发展，中国乃至世界物流经历了连续十多年的爆炸式增长。例如，2021 年菜鸟、京东、顺丰等大型物流公司的全国快递业务量突破 1083 亿件，随着构建新发展格局的加快和物流需求的增长，未来我国物流业务量仍会保持较快的增长。同时，物流行业的高速发展对大型实时物流调度系统提出了更高的要求。然而，物流配送产生的运输和仓储成本居高不下。基于物流配送现状和时代需求，寻求高效的物流配送模式受到了学界和业界的广泛关注。多车场车辆路径问题(Multi-Depot Vehicle Routing Problem, MDVRP)具有广泛的应用场景，包括交通运输、物流配送和快递分发等实际情况，探索该问题的高效求解方法对我国供应链发展具有重要的理论和现实意义。MDVRP 问题属于车辆路径问题(Capacitated Vehicle Routing Problem, CVRP)的一个变体。由于 CVRP 已是 NP-hard 问题，而相比单车场的 CVRP 而言，MDVRP 的解空间更加庞大。因此，MDVRP 也属于 NP-hard 问题。

求解 MDVRP 的传统方法主要包括精确算法、多项式时间近似算法、元启发式算法。精确算法能够求得最优解，但由于其 NP-hard 性质很难应用于求解 50 个客户以上的问题[1]。多项式时间近似算法通常能够得到有质量保证的解，但最优性保证较弱，甚至不能得到该问题的局部最优解。元启发式算法，例如狼群算法[2]、蚁群优化算法[3]、蝙蝠算法[4]和变邻域搜索算法[5]，由于其高性能被广泛使用，但通常需要针对特定的问题定制和专业的领域知识[6]，并且难以在多项式时间内寻找到大规模问题的较优解。以上三种方法很少利用优化问题的共同特征，经常反复求解相同类型问题的算例，对于这些算例可以认为目标函数或约束中的系数值是从相同的基础分布中采样所得[7]。尽管出现了大量的求解策略，

收稿日期：2022-03-07；**修回日期：**2022-04-29　**基金项目：**浙江省高校重大人文社科攻关计划资助项目(2018QN060)

作者简介：雷坤(1998-)，男，四川南充人，硕士研究生，主要研究方向为深度强化学习、运筹优化；郭鹏(1988-)，男，四川南充人，副教授，硕导，博士，主要研究方向为生产调度、智能制造、运作管理、智慧物流、深度强化学习等；王祺欣(1999-)，男，四川南充人，硕士研究生，主要研究方向为深度强化学习、路径优化；赵文超(1996-)，男，重庆巫溪人，硕士研究生，主要研究方向为生产调度；唐连生(1974-)，男(满族)(通信作者)，辽宁盖州人，教授，副院长，博士，主要研究方向为数字物流与智能技术、物流产业经济、物流系统仿真、港口物流等（lianshengtang_nb@163.com）.

但求解效率仍然有进一步的提升空间和基于更加高效的求解框架搭建的必要。因此，引入学习的方法以高效寻找接近最优解决方案尤为重要。

近年来，越来越多的研究将深度强化学习(Deep Reinforcement Learning, DRL)技术应用于求解组合优化问题，并取得了突破性进展。表 1 对现有基于强化学习求解路径问题的方法进行了总结。强化学习可以进一步分为基于模型的和无模型的方法。而无模型强化学习方法可以分为 Value-based 和 Policy-based 的方法或者两者的结合(Actor-critic)。此外，按路径问题可以分为旅行商问题(Travelling salesman problem, TSP)、CVRP 和 MDVRP。

表 1 求解路径问题的深度/强化学习方法汇总

Tab. 1　Survey of deep/reinforcement learning methods in solving routing problems

路径问题	文献	网络结构	方法类型
TSP	文献[8]	Transformer	无模型的 RL, Actor-Critic
	文献[9]	NN	无模型的 RL, Policy-Based
	文献[10]	GPN	无模型的分层 RL, Policy-Based
	文献[11]	GAT	无模型的 RL, Actor-Critic
	文献[12]	GCN	无模型的 RL, Given Model
	文献[13]	GAT+注意力机制	无模型的 RL, Actor-Critic
	文献[14]	Transformer	无模型的 RL, Policy-Based
	文献[15]	GNN	无模型的 RL, Policy-Based
CVRP	文献[16]	LSTM+注意力机制	无模型的 RL , Actor-Critic
	文献[17]	LSTM	无模型的 RL, Actor-Critic
	文献[8]	Transformer	无模型的 RL, Actor-Critic
	文献[18]	LSTM+注意力机制	无模型的 RL, Actor-Critic
	文献[19]	NN	无模型的 RL, Actor-Critic
	文献[20]	GAT+GRU	无模型的 RL, Actor-Critic
	文献[13]	GAT+注意力机制	无模型的 RL, Actor-Critic
	文献[21]	GAT+GRU	无模型的 RL, Actor-Critic
	文献[22]	Transformer	无模型的 RL, Actor-Critic
MDVRP	文献[23]	Transformer	无模型的 RL, Actor-Critic
	本文	**RE-GAT+Transformer**	**无模型的 RL, Actor-Critic**

注：在网络结构列，LSTM: long short-term memory; NN: neural networks; GRU: gate recurrent unit; GPN: graph pointer network.

大多数基于 DRL 的应用集中在路径问题，如 TSP 和 VRP。Vinyals 等[24]引入了 sequence-to-sequence 模型指针网络(PtrNet)的监督学习框架来训练求解 TSP 等组合优化问题，该模型通过 Softmax 注意力机制(指针)来选择输入序列中的元素作为输出的递归架构。Bello 等[25]引入了 Actor-Critic 风格的深度强化学习算法以无监督的方式训练 PtrNet 来求解 TSP，并且其性能在多达 100 个节点的 TSP 上优于以前的大多数近似算法。Nazari 等[16]在 Bello 的框架上进行了扩展以解决 VRP。

包括 VRP 在内的大多数组合优化问题都具有图结构[6]，可以很容易地通过现有的图嵌入或图网络嵌入技术来建模，将图信息嵌入到连续的节点表示中。图神经网络的最新发展可以用于网络设计，因为它在信息嵌入和图拓扑的信念传播方面具有很强的能力[6]。然而，上述工作中使用的 sequence-to-sequence 神经网络结构不能充分利用并提取该问题的图结构信息，例如图中节点包含客户的位置和需求信息、边包含权重信息。作为处理非欧氏数据和捕捉图结构信息的有力工具，图神经网络(Graph Neural Network, GNN)近年来得到了广泛的研究。

近年来基于 GNN 的近似求解器经过训练后，其算法时间复杂度明显优于传统的运筹优化算法。Li 等[26]应用图卷积网络(GCN)模型[27]以及引导树搜索算法来解决基于图的组合优化问题，如最大独立集和最小顶点覆盖问题。Dai 等[7]通过 GNN 对问题算例进行编码，与序列到序列模型相比，图神经网络具有节点顺序不变性，更好地反映了 TSP 的组合结构，他们使用 DQN[28]训练 structure2vec 图嵌入模型[29]。受 Transformer 架构[30]的激励，Kool 等[8]提出了注意力模型以解决多种组合优化问题，并在策略梯度算法中使用 Rollout 基线显著的改善了小规模路径问题的求解结果。Nowak 等[31]以监督学习的方式使用深度 GCN 通过高度并行的波束搜索以非自回归的方法构建有效的 TSP 图表示并输出行程。Drori 等[13]开发了新的框架来解决图上的组合优化问题，该框架运用(Graph Attention Networks, GAT)求解众多图组合优化问题，他们称该框架具有从小图上的训练到大图上的测试和从随机图上的训练到在现实世界的图上测试的泛化性能。

然而大多数机器学习方法聚焦于求解单车场的车辆路径问题，对于多车场车辆路径问题的研究较少。王万良等[23]基于多头注意力机制设计了多智能体强化学习框架求解 MDVRP，并利用策略梯度算法进行训练，他们的实验结果表明所提出的多智能体深度强化学习模型及其与搜索策略的结合能够快速获得高质量的解。然而，文献[23]没有验证其训练后的模型泛化到不同规模算例的性能，及泛化到真实世界算例(标准算例)的性能。相反，本文提出的编码器-解码器模型不受问题规模(即车场和客户数)的约束，即经过训练的模型可适用于任意车场和客户数的算例，且能够在毫秒级给出解决方案。该框架具有较强的泛化性能。通过对随机生成数据集的测试，验证了该框架的有效性。此外，通过 VRPLIB 的标准算例测实验实验证了该框架具有从随机算例训练到真实世界算例测试的泛化能力。

以上基于 GNN 的 Learning-based 方法激励本文探索其在求解 MDVRP 中的潜力。提出了基于端到端(End-to-end)的深度强化学习框架用于高效求解 MDVRP。在该框架中，首先将 MDVRP 建模为马尔可夫决策过程(Markov Decision Process, MDP)。本文提出了残差-边-图注意力网络(Residual edge graph attention network, RE-GAT)模型作为编码器提取嵌入 MDVRP 图表示的状态特征，该模型是对图注意力网络(graph attention network, GAT)的改进。GAT 在提取图结构信息的过程中仅考虑节点的信息忽视了边的信息，而边的特征可以为学习策略提供与优化目标相关的更多直接信息(如加权距离)。此外，同时输入节点和边信息有利于挖掘不同节点之间空间邻接关系的特征。提出的 RE-GAT 模型将 MDVRP 图表示中节点和边(如权重)的信息进行融合并更新，并在层与层之间添加了残差连接有效地防止了深层模型中梯度消失和模型退化的问题。此外，还基于 Transformer 模型设计了解码器用于求解过程中高效地预测节点。所提出的编码器-解码器模型一经训练即可适用于任意车场和客户数量的算例，且能够在毫秒级给出路径优化方案。换言之，该框架可作为一种线下训练、线上测试的实时优化框架。

为验证该框架的可行性和有效性，本文设计了随机生成的算例对框架进行训练和测试。此外，通过 VRPLIB 标准算例测试该框架的泛化性能，并与最新的基于机器学习方法和元启发式算法进行对比以验证所提框架的优越性。最后，通过实验验证并分析了框架在训练和测试阶段运行时间的复杂度。

1　问题描述

一般地，MDVRP 可以描述为具有容量限制的一辆车向需求有限的多个客户运送货物，当车辆载物用完或不满足客户需求时返回仓库,其目标是在满足所有客户需求的基础上，使得总路线长度最小化，图 1 展示了具有两车场的 MDVRP

示意图。本文通过无向图 $G=(V,E,W)$ 来定义 MDVRP，其中节点包括客户和多个车场 $i=\{1,\ldots k,k+1,\ldots,m\}$ 表示其原始特征 n_i，包括该节点的坐标 n_i' 和需求信息 δ_i。其中，$i=\{1,\ldots,k\}$ 表示车场节点，$i>k$ 表示第 i 个客户节点。客户节点 $i,i\in\{k+1,\ldots,m\}$ 的货物需求为 δ_i，其中 $0<\delta_i<D$，且 $D>0$ 表示车辆的容量。假设车场的需求 $\delta_i=0,\ i\in\{1,\ldots,k\}$。$a_{ij}\in E,i,j\in V,i\neq j$ 表示从节点 i 到节点 j 的边，$e_{ij}\in W$ 表示 a_{ij} 的距离信息。车场和客户节点坐标均从单位平方 $[0,1]\times[0,1]$ 中随机生成，即对客户节点数量为 20、30、50 的问题，本文分别生成 $n=20+k、50+k、100+k$ 个节点，与之三个规模问题相对应的车辆容量为 30、40、50，每个客户节点需求在 $\{1,\cdots,9\}$ 中随机产生。此外，将客户节点需求归一化到 $[0,1]$ 之间，车辆容量 D 相应地变换为 3、4、5。

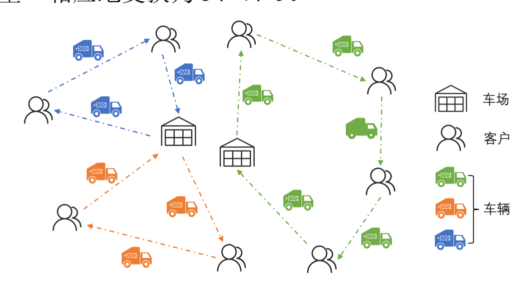


图 1　MDVRP 示意图

Fig. 1　The description of MDVRP

本文引入所有节点的排列 $\hat{\pi}=(\hat{\pi}_1,\ldots,\hat{\pi}_z)$ 表示该问题的解，其中 $\hat{\pi}_t\in\{1,\ldots,z\}$ 并且 $\hat{\pi}_t\neq\hat{\pi}_{t'}$，$\forall t\neq t'$。本文的目标是在给定问题算例的情况下找到问题的解 $\hat{\pi}$，使得每个客户节点只能访问一次(车场节点可以多次访问，即有 $z\geq m$)，并且总的路线长度最小。排列 $\hat{\pi}$ 的长度定义为

$$L(\hat{\pi}|s)=\|n_{\hat{\pi}_z}-n_{\hat{\pi}_1}\|_2+\sum_{t=1}^{z-1}\|n_{\hat{\pi}_t}-n_{\hat{\pi}_{t+1}}\|_2,\tag{1}$$

其中 $\|\cdot\|_2$ 表示 2 范数，$\|n_{\hat{\pi}_z}-n_{\hat{\pi}_1}\|_2$ 表示最后服务的客户节点到车场的距离。本文的图-注意力模型为 MDVRP 算例 s 定义了一个随机策略 $p(\hat{\pi}|s)$。基于链式概率法则，序列 $\hat{\pi}$ 的选择概率可以基于图-注意模型的参数集合 θ 计算:

$$p_\theta(\hat{\pi}|s)=\prod_{t=1}^{m}p_\theta(\hat{\pi}_t|s,\hat{\pi}_{t'},\ \forall t'<t).\tag{2}$$

编码器对所有输入节点的原始特征编码嵌入到高维特征。解码器基于编码器的输出在每一个时间步骤 t 选择一个节点，从而产生输入节点的排列 $\hat{\pi}$。

2　MDVRP 的马尔可夫决策过程定义

本节对 MDVRP 的马尔可夫决策过程(Markov decision process, MDP)进行建模，其中包括对状态、动作和收益的定义:

a) **状态**：在时间步骤 t，状态由已访问的节点所构成的子图 $G'(G'\subseteq G=(V,E,W))$ 表示。

b) **动作**：在时间步骤 t，动作是未进行服务的客户节点或车场节点。

c) **收益**：本文的优化目标是最小化车辆的行驶距离 $L(\hat{\pi}|s)$。首先计算时间步骤 t 到时间步骤 $t+1$ 所访问的两个节点的距离($\|n_{\hat{\pi}_t}-n_{\hat{\pi}_{t+1}}\|_2$)，然后将智能体的即时收益定义为: $-\|n_{\hat{\pi}_t}-n_{\hat{\pi}_{t+1}}\|_2$(强化学习的目标是最大化收益，因此取负值)。

3　参数化强化学习智能体行为策略网络

3.1　编码器

编码器将图 $G=(V,E,W)$ 作为输入，其结构如图 2 所示。输入节点特征为 n_i，输入边的特征为欧氏距离 $e_{ij},i,j\in\{1,\ldots,m\}$。上述两个特征分别通过全连接的层(图 2 中的 FC 层)嵌入(embedding)到 d_x 和 d_e 维特征中，然后被送入 RE-GAT 进行编码。式(3)和(4)分别描述了节点和边的嵌入过程。

$$x_i^{(0)}=BN(A_0n_i+b_0),i\in\{1,\ldots,m\}\tag{3}$$

$$\hat{e}_{ij}=BN(A_1e_{ij}+b_1),\ i,j\in\{1,\ldots,m\}\tag{4}$$

其中: A_0 和 A_1 分别表示可学习的权重矩阵，b_0 和 b_1 分别表示可学习的权重向量，$BN(\cdot)$ 表示批归一化(batch normalization)[32]。

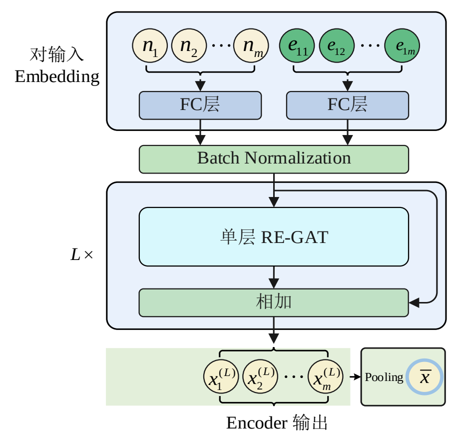


图 2　编码器整体结构

Fig. 2　Encoder network structure

本文中编码器包含 L 层 RE-GAT，图 3 描述了单层 RE-GAT 如何将边的信息集成到节点信息中并更新每个节点的信息。注意力系数 α_{ij}^{ℓ} 表示第 $l\ (l\in\{1,\ldots,L\})$ 层中节点 j 相对于节点 i 的权重系数(注意力系数)，其中 $i,j\in\{1,\ldots,m\}$:

$$\alpha_{ij}^{\ell}=\frac{\exp\left(\sigma\left(g^{\ell T}\left[W^{\ell}\left(x_i^{(\ell-1)}\,\|\,x_j^{(\ell-1)}\,\|\,\hat{e}_{ij}\right)\right]\right)\right)}{\sum_{z=1}^{m}\exp\left(\sigma\left(g^{\ell T}\left[W^{\ell}\left(x_i^{(\ell-1)}\,\|\,x_z^{(\ell-1)}\,\|\,\hat{e}_{iz}\right)\right]\right)\right)}\tag{5}$$

其中 $(\cdot)^T$ 表示转置运算符，$\cdot\|\cdot$ 表示连接操作符，g^{ℓ} 和 W^{ℓ} 是可学习的权重向量和权重矩阵，$\sigma(\cdot)$ 是 LeakyReLU 激活函数。图 2 展示了具有多层 RE-GAT 的编码器模型结构。RE-GAT 的每一层通过式(5)和(7)描述的注意机制更新每个节点的特征向量。模型在每两层之间使用了残差连接(由式(6)表示)。也就是说，第 ℓ 层的输出被计算为

$$x_i^{(\ell)}=x_{i,R}^{(\ell)}+x_i^{(\ell-1)},2<\ell<L,\tag{6}$$

其中 $x_{i,R}^{(\ell)}$ 由式(7)计算所得。

$$x_{i,R}^{(\ell)}=\sum_{j=1}^{m}\alpha_{ij}^{\ell}W_1^{\ell}x_j^{(\ell-1)},\tag{7}$$

其中: W_1^{ℓ} 表示可学习的权重矩阵。第 L 层 RE-GAT 输出每个节点的最终嵌入特征向量 $x_i^{(L)}$。然后，用它们计算最终的图嵌入向量 $\bar{x}=\{\bar{x}_1,\ldots,\bar{x}_{d_x}\},\bar{x}_j\in\mathbb{R}$，对于每个节点 $j\in\{1,2,\cdots,d_x\}$ 由式(8)表示:

$$\bar{x}_j=\frac{1}{m}\sum_{i=1}^{m}(x_i^{(L)})_j,\ j=1,\cdots,d_x\tag{8}$$

3.2　解码器

在解码过程中，解码器基于注意力机制生成待选择节点(所有车场和客户)的概率分布，即每个节点都会关联一个概率值。然后，通过掩码(下文将详细介绍)机制来处理相关约束，即避免重复访问已服务的客户节点和连续两次选择车场节点。最后，搜索策略基于所输出的概率分布进行节点选择，如贪婪搜索(贪婪地选择概率最大的节点)或采样的解码策略(基于概率分布进行采样)。调度中心的选取同样适用该解码机制。传统的启发式算法通常采用“先分组后规划”的思想[23]求解 MDVRP，存在以下缺点[23]:a)不同分组各自规划,这导致分组之间整体关联性的缺失;b)启发式方法中分组的优劣通常决定了整体规划的优劣,而分组规则的制定需要专家领域知识，人为选取的分组规则很难达到最优效果。相反，深度

强化学习智能体能够通过数据驱动的方式与调度环境进行交互，进而不断更新进化自身策略以最大化收益(对于路径问题为总路线长度的负值)。即在本文的解码过程中强化学习策略选取调度中心的过程中无须无须依靠人为启发式地进行干预。

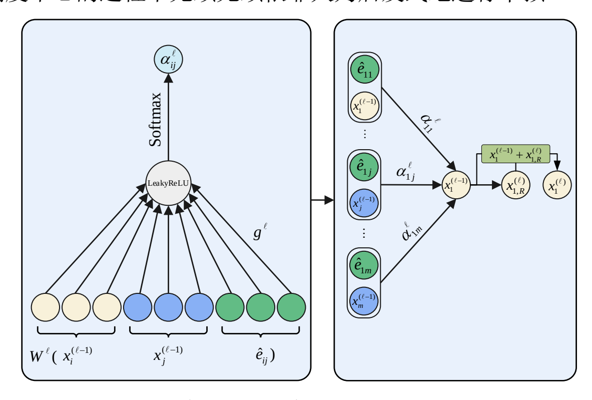


图 3 边和节点信息的聚合和更新方式

Fig. 3 Edge and node fusion and update method

本文采用类似于 Transformer[30]模型所采用的多头注意力机制来设计针对 MDVRP 的解码器。与原始 Transformer 模型的结构不同，本文所设计的基于多头注意力机制的解码器为了提升计算效率只包含两个注意力子层，且不使用残差连接、批量归一化和全连接层网络。第一层通过多头注意机制计算上下文向量，第二层输出所选节点的概率分布，并基于该分别选择节点。

解码按顺序进行，在时间步骤 t，首先利用图嵌入向量 $\bar{x}$、$t-1$ 时刻选择的节点 $\hat{\pi}_{t-1}$ 的嵌入向量和车辆的剩余容量计算出上下文向量 $c_t^{(0)}$：

$$c_t^{(0)} = \begin{cases} \bar{x} + W_x(x_{\hat{\pi}_{t-1}}^{(L)} \parallel D'_{t-1}), t>1 \\ \bar{x} + W_x(x_0^{(L)} \parallel D'_t) \quad ,t=1 \end{cases}. \tag{9}$$

其中：W_x 是可学习的权重矩阵，D'_{t-1} 和 D'_t 分别表示两个解码步骤车辆的剩余容量。其中 D'_t 的更新公式如下：

$$D'_t = \begin{cases} D'_{t-1} - \delta'_{\hat{\pi}_t} & ,\hat{\pi}_t \in \{1,\ldots,m\} \\ D & ,\hat{\pi}_t = 0 \end{cases}. \tag{10}$$

解码器第一层的输入是上下文向量 $c_t^{(0)}$，该层产生新的上下文向量 $c_t^{(1)}$。特别地，该上下文向量是通过一个多头(H 头)注意力机制获得的。式(11)描述了多头注意力机制，通过编码器输出的节点的嵌入向量和上下文向量 $c_t^{(0)}$ 来分别计算键向量 $k_i \in \mathbb{R}^{d_v}$，值向量 $v_i \in \mathbb{R}^{d_v}$，查询向量 $q \in \mathbb{R}^{d_v}$：

$$q = W^Q c_t^{(0)}, \quad v_i = W^V x_i^{(L)}, \quad k_i = W^K x_i^{(L)}, \\ i \in \{1,2,\cdots,m\}. \tag{11}$$

其中：$W^K \in \mathbb{R}^{d_v \times d_x}$，$W^Q \in \mathbb{R}^{d_v \times d_x}$ 和 $W^V \in \mathbb{R}^{d_v \times d_x}$ $(d_v = d_x / H)$ 都是可学习的权重矩阵。

本文使用上下文向量 $c_t^{(0)}$ 来计算每个查询向量 q。然后，利用编码器输出的节点嵌入向量 $x_i^{(L)}$，$i \in \{1,2,\cdots,m\}$ 计算键向量 $k = \{k_1,\cdots,k_m\}$ 和值向量 $v = \{v_1,\cdots,v_m\}$。并利用查询向量 q 和键向量 $k = \{k_1,\cdots,k_m\}$ 计算第一解码层的注意系数 $u_{i,t}^{(1)} \in \mathbb{R}, i \in \{1,\cdots,m\}$：

$$u_{i,t}^{(1)} = \begin{cases} -\infty, if \;\; i \neq \hat{\pi}_{t'} \, (\forall t' < t) or \;\; \delta'_i > D'_{t-1} \\ \quad or \, t=1 \; or \;\; \hat{\pi}_{t-1} \in \{1,\ldots k\} \\ \dfrac{q^T k_i}{\sqrt{d_v}}, \quad otherwise. \end{cases} \tag{12}$$

本文通过掩码(*mask*)来避免重复选择客户节点、车辆剩余容量不满足该客户节点和连续两次选择车场节点(即在 $t-1$ 时刻选择车场节点 $\hat{\pi}_{t-1} \in \{1,\ldots k\}$)。具体地，在式(12)中，通过将上述情况的注意力系数设置为 $-\infty$ 来掩码。然后使用式(13)通过 Softmax 激活函数将注意力系数 $u_{i,t}^{(1)}$ 归一化：

$$\hat{u}_{i,t}^{(1)} = softmax\left(u_{i,t}^{(1)}\right), i \in \{1,\cdots,m\} \tag{13}$$

其次，根据式(14)计算第 h ($h \in \{1,\cdots,H\}$ 头归一化注意力系数 $(\hat{u}_{i,t}^{(1)})^h$，其中 $1 \le i \le m$。然后将每个头计算出的向量串联起来，并通过全连接层计算得到最终的上下文向量 $c_t^{(1)}$：

$$c_t^{(1)} = W_f \cdot \left(\Vert_{h=1}^{H} \sum_{i=1}^{m} (\hat{u}_{i,t}^{(1)})^h v_i^h \right), \tag{14}$$

其中：W_f 是可学习的权重矩阵。该多头注意机制有助于提高注意力学习过程的稳定性[33]。

第二层解码器基于单头注意力机制，其输入是上下文向量 $c_t^{(1)}$。然后利用式(15)计算第二解码层在 t 时刻的注意系数 $u_{i,t}^{(2)} \in \mathbb{R}, i \in \{1,\ldots m\}$。基于 Bello 等的工作[25]，采用 tanh 激活函数将该系数截断在 $[-C,C]$ 内(本文选取 $C=10$)。然后通过式(16)采用 Softmax 激活函数获得每个节点的选择概率 $p_{i,t}$，$i \in \{1,\ldots m\}$：

$$u_{i,t}^{(2)} = \begin{cases} -\infty, if \;\; i \neq \hat{\pi}_{t'} \, (\forall t' < t) or \;\; \delta'_i > D'_{t-1} \;\; or \\ \quad t=1 \; or \;\; \hat{\pi}_{t-1} \in \{1,\ldots k\} \\ C \cdot \tanh\left(\dfrac{c_t^{(1)T} k_i}{\sqrt{d_v}}\right) \quad ,otherwise \end{cases}, \tag{15}$$

$$p_{i,t} = p_\theta(\hat{\pi}_t | s, \hat{\pi}_{t'}, \forall t' < t) = softmax\left(u_{i,t}^{(2)}\right). \tag{16}$$

最后，根据策略概率分布的 $p_{i,t}$，使用采样或贪婪解码(将在下文介绍)来预测下一个要访问的节点(车场或客户节点)。

4 基于策略梯度的深度强化学习算法

本文引入改进的 REINFORCE 算法来训练所提出的模型。将损失函数定义为 $Loss(\theta|s) = \mathbb{E}_{\hat{\pi} \sim p_\theta(\hat{\pi}|s)}[L(\hat{\pi}|s)]$。该改进 REINFORCE 算法具有 actor-critic 风格，但又与传统地将状态-价值估计函数作为 critic 不同。与该算法的原始版本相比，本文所实现的版本添加了 Rollout 基线方法[8]，以加速算法收敛速度以及增强优化性能。其损失函数的梯度计算过程如下：

$$\nabla_\theta Loss(\theta|s) = \mathbb{E}_{\hat{\pi} \sim p_\theta(\hat{\pi}|s)}[(L(\hat{\pi}|s) - b)\nabla_\theta log p_\theta(\hat{\pi}|s)]. \tag{17}$$

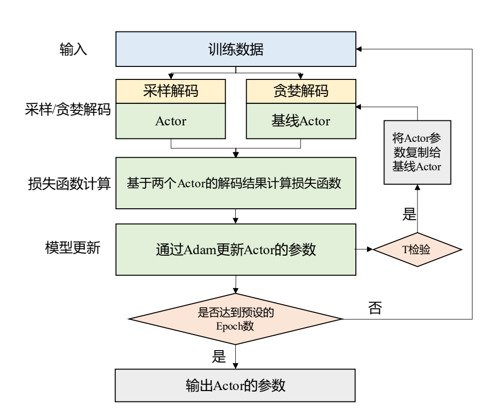


图 4 改进 REINFORCE 算法流程框图

Fig. 4 Improvement reinforce algorithm process block diagram

在所提出的具有 Rollout 基线版本的 REINFORCE 算法中，critic 网络被基线 actor 所取代。该算法的流程框图如图 4 所示。算法可以描述为具有两个 actor 的结构，基线 actor 的策略网络 $\pi_{\theta^{BL}}$ (θ^{BL} 为参数集合)在每个 epoch 内被固定(即其参数不进行更新)，该策略类似于 DDQN[34]中固定目标 Q-网络。在每个 epoch 结束时，使用贪婪解码来比较当前训练 actor 和基线 actor 的结果。然后，基线 actor 策略网络的参数只有在测试算例上具有显著提升(对其进行 t 检验，显著性水平 $\alpha = 5\%$)才会进行更新。在训练过程中，本文还采用“代码级优化”的策略对该算法进行改进，包括 Adam 优化器的学习率衰减和奖励函数的归一化，提高了该算法的性能。

有效的组合优化搜索算法主要包括束波搜索、邻域搜索和树搜索。Bello 等[25]提出了诸如采样、贪婪搜索和主动搜索等搜索策略。本文使用了以下两种解码策略。

a) 贪婪解码：一般来说，贪婪算法构造局部最优解并提供全局最优解的快速近似值。在每个解码步骤中，贪婪地选择概率最高的节点，当所有节点的要求都被满足时，即搜索终止，从而构造出有效解。

b) 随机采样：在每个解码时间步骤 t，随机策略 $p_\theta(\hat{\pi}_t|s,\hat{\pi}_{t'}\ \forall t'<t)$ 根据概率分布随机选择节点来构造有效解。在测试过程中，Bello 等[25]利用温度超参数 $\lambda\in\mathbb{R}$ 对式(16)进行修正，以保证采样的多样性。修改后的公式如下：

$$p_{i,t}=p_\theta(\hat{\pi}_t|s,\hat{\pi}_{t'}\ \forall t'<t)=softmax\left(\frac{u_{i,t}^{(2)}}{\lambda}\right). \tag{18}$$

通过对温度超参数的网格搜索，发现温度值分别为 2.5、1.8、1.2 时对于 MDVRP20(客户节点数为 20)、MDVRP50 和 MDVRP100 效果最好。

在训练过程中，通常需要模型探索环境以获得更好的模型性能，因此采用随机采样的解码策略。而在测试过程中，本文使用了贪婪解码策略。此外，根据现有的研究的测试方法[8][25]，本文还采用随机采样的方法求得了 1280 个解决方案并报告其最好的一个。

5　计算实验

本节通过实验验证所提出框架的可行性和有效性。实验包括训练和测试两个部分，由于训练需要大量数据，因此训练数据由均匀分布随机产生。测试数据集包括随机生成的算例和公开的标准算例(Cordeau 等提出[35, 36]) 其分别用于测试所提出框架的有效性和泛化性能。此外本文还将所提出的框架与其他 Learning-based 的方法、Google OR-tools 和元启发式算法进行了比较。

5.1　数据集与超参数的选择

为提高可读性，本文将 MDVRP 的规模以“客户数-车场数”的格式表示。本文所用到的数据集包括随机生成的训练数据、验证数据和随机测试数据。分别针对规模 20-2、50-2 和 100-2 从单位平方 $[0,1]\times[0,1]$ 中随机生成 MDVRP 算例。并分别为 20-2 的训练集生成了 819 200 个算例，为 50-2 和 100-2 的训练集生成了 768 000 个算例，且每个模型训练 100 个 Epoch。对于验证数据和随机测试数据，使用与训练数据相同的分布，分别为每个规模生成 10 000 个算例。此外，本文还采用公开的标准算例(Cordeau 等提出[35, 36])来评估所提出的模型由随机生成的算例训练到真实世界算例测试的泛化性能以及测试不同规模算例(不同数量的车场以及客户)的泛化性能。所有实验在具有一块 Turbo HT (100W) DDR4-2400 CPU 和一块 Nvidia GeForce RTX 3090 GPU 的计算机上进行。表 2 列出了训练过程的其他相关超参数的值。本文的模型由 PyTorch 构造，并用 Python 3.7 进行实现。

表 2　超参数选值

Tab. 2　Value of hyper-parameters

超参数	值
编码器层数 L	4
学习率衰减系数 β	0.96
学习率 l	$1\times10^{-3}\,(m=20)$ $3\times10^{-4}\,(m=50,100)$
多头注意力机制头数 H	8
节点的信息嵌入维数 d_x	128
边的信息嵌入维数 d_e	16
优化器	Adam[37]

5.2　计算结果分析

5.2.1 随机算例分析

表 3 列出了所提框架(根据解码策略的不同由“Greedy”、“Sampling128”和“Sampling1280”表示)、Google OR-tools 和其他基于 DRL 的方法[23]在不同规模随机生成的 MDVRP 上的测试结果。该表中距离(越小越好)和相对最优 Gap 值为 10 000 个算例的平均值。此外，还给出了所有测试算例的平均运算时间。

表 3　所提出的框架、其他强化学习方法和 Google OR-tools 的随机算例计算结果

Tab. 3　Results of the proposed framework, a reinforcement learning method and Google OR-tools on random generated MDVRP instances

方法	类型	MDVRP20-2			MDVRP50-2			MDVRP100-2		
		距离	Gap 值	时间	距离	Gap 值	时间	距离	Gap 值	时间
Greedy[23]	RL, G		-			-		15.62	13.68%	43.9 ms
Greedy (本文)	**RL, G**	**5.35**	**1.71%**	**0.2 ms**	**9.40**	**4.56%**	**0.7 ms**	**14.46**	**5.24%**	**1.8 ms**
OR-tools	H, S	5.65	7.41%	0.18 ms	9.73	8.23%	1.02 ms	14.99	9.10%	3.6 ms
Sampling128[23]	RL, S		-			-		15.07	9.68%	5.68 s
Sampling128 (本文)	**RL, S**	**5.33**	**1.33%**	**19 ms**	**9.27**	**3.12%**	**0.07 s**	**14.34**	**4.36%**	**0.21 s**
Sampling1280 (本文)	**RL, S**	**5.26**	**0.00%**	**91 ms**	**8.99**	**0.00%**	**0.46 s**	**13.74**	**0.00%**	**1.58 s**

由该表可以看出，采样解码策略能够获得所有方法给出结果的最好解。该策略对每个算例都执行 128 或 1280 次采样，即构造 128 或 1280 个解并报告最好的一个。相反，贪婪解码策略仅通过训练后的模型在每次解码过程中贪婪的选择具有最高概率的节点以构造单个解。此外，基于神经网络的并行计算可以对多个算例进行批处理，这使得训练后的模型以贪婪解码的方式具有极快的求解速度。例如，对于规模为 100-2 的 10 000 个 MDVRP 算例，所提出的方法以贪婪解码的方式求解单个算例只需要 1.8 ms，而采样的解码方式需要 1.58 s。

Google OR-tools 是基于局部搜索的高效求解器，文献[23]是深度强化学习方法。然而，在所有规模的 MDVRP 问题上，所提出的框架无论采用贪婪解码还是采样解码的方式都优于 OR-tools 和文献[23]所提出强化学习方法，且贪婪解码方式在运行时间上也要远优于这两种方法。此外，图 5 展示了训练过程中各规模 MDVRP 的收敛曲线，可以看出各规模的算例训练过程中在 80 个 Epoch 后都能很好地收敛。

5.2.2 公开算例分析

为评估所提出框架从随机生成的算例泛化到真实世界算例以及泛化到不同规模(即不同客户和车场数)的性能，本节将训练后的模型(通过随机生成的规模为 100-2 的 MDVRP 算例进行训练)用于求解 Cordeau 等[35, 36]针对 MDVRP 提出的公开标准算例(客户数 50-160)，所有结果均列在表 4 中。此外，为进一步评估所提出框架的性能，还与最先进的元启发式算法(改进 ACO[38])进行比较，其结果同样列在表 4 中。两种方法的 Gap 值均基于已知最好的解(best known solutions, BKS)求得。

由表 4 可以看出，虽然改进 ACO 算法在大部分算例上优于本文所提出的框架，且其平均 Gap 值也更优(1.98% vs. 4.97%)。但该改进 ACO 方法的结果是针对每个算例单独执

行 100 次并报告最好的一个，表中时间为平均时间。相反，本文所提出的框架采用贪婪解码的策略对每个算例进行单次求解。本文所提出的框架在运行时间上要远远优于该改进 ACO 算法[38](0.09 s vs. 51.59 s)。因此，本文所提出的框架在该数据集的求解质量和运行时间上较为均衡。此外，所提出的框架可以作为一种线下训练线上测试使用的实时求解框架。

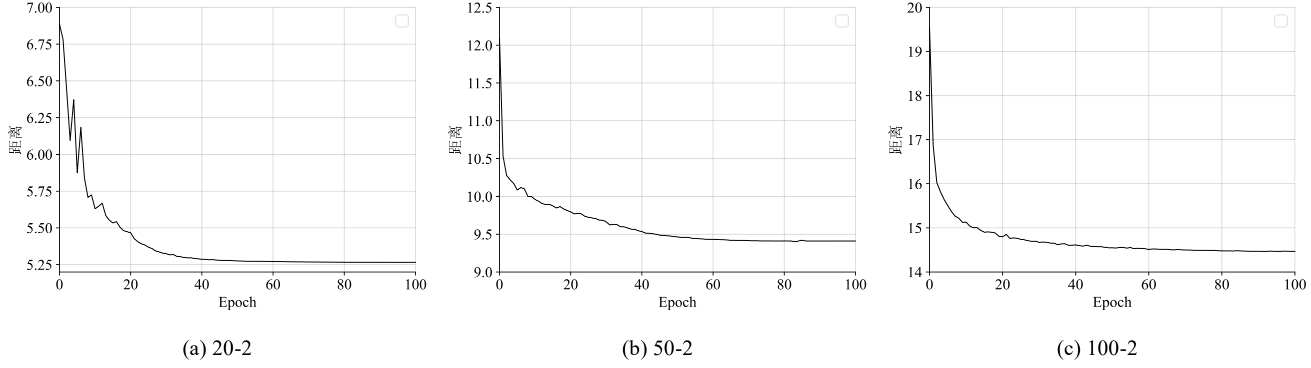


图 5 各规模 MDVRP 训练过程收敛曲线

Fig.5 Convergence curve of MDVRP with different scales in training process

5.2.3 拓展计算分析(CVRP)

为进一步评估所提出框架对于不同场景车辆路径问题的求解性能及泛化能力，将所提出的框架用于求解单车场的 CVRP。本节采用随机算例进行实验验证，即由均匀分布随机生成 10 000 个算例(与文献[8][16]保持一致)。并通过与现有的 Learning-based 方法(表 5 种“PtrNet”和“AM”都是基于深度强化学习的方法)、Google OR-tools、Gurobi 精确求解器和 LKH3 求解器进行对比，来验证所提出框架在单车场 CVRP 上的有效性。不同规模的 CVRP 算例的测试结果列于表 5 中。表中所列出的结果根据其求解方法的属性可以分为三个大类，包括求解器、贪婪方法和采样/搜索方法。除了所提出模型的结果，该表的其他结果均取自 Kool 等[8]。在模型性能方面，所提出的框架无论是采样还是贪婪解码策略的结果都要优于其他所列出的基于学习的方法。其结果说明了所提出的框架在单车场 CVRP 场景下仍然具有较好的泛化性能。同时，验证了该模型具有迁移到其他场景 VRP 的潜力。

表 4 所提出方法和改进 ACO 的结果

Tab. 4 Results of the proposed framework and improved ACO on benchmarks

No.	问题	客户数	车场数	BSK	Greedy			改进 ACO[38]		
					距离	Gap 值	时间	距离	Gap 值	平均时间
1	p01	50	4	576.87	633.47	9.81%	**0.06 s**	**607.66**	**5.34%**	1.7 s
2	p02	50	4	473.53	**493.34**	**4.18%**	**0.06 s**	495.34	4.61%	1.8 s
3	p03	75	5	641.19	683.71	6.63%	**0.09 s**	**670.82**	**4.62%**	4.3 s
4	p04	100	2	1001.59	1104.65	10.28%	**0.10 s**	**1021.36**	**1.97%**	28.4 s
5	p05	100	2	750.03	809.79	7.96%	**0.10 s**	**750.72**	**0.09%**	25.6 s
6	p06	100	3	876.50	961.77	9.72%	**0.09 s**	**902.91**	**3.01%**	31.9 s
7	p07	100	4	885.80	980.08	10.64%	**0.09 s**	**907.55**	**2.46%**	30.9 s
8	p12	80	2	1318.95	1329.82	0.82%	**0.04 s**	**1318.95**	**0.00%**	15.4 s
9	p13	80	2	1318.95	1329.82	0.82%	**0.04 s**	**1318.95**	**0.00%**	16.0 s
10	p14	80	2	1360.12	**1360.12**	**0.00%**	**0.04 s**	1365.69	0.41%	16.9 s
11	p15	160	4	2505.42	2586.66	3.24%	**0.15 s**	**2554.12**	**1.94%**	167.1 s
12	p16	160	4	2572.23	**2586.66**	**0.56%**	**0.15 s**	2606.22	1.32%	188.1 s
13	p17	160	4	2709.09	**2709.09**	**0.00%**	**0.15 s**	**2709.09**	**0.00%**	147.3 s
平均值						4.97%	**0.09 s**		**1.98%**	51.59 s

表 5 不同方法在各规模 CVRP 上的结果

Tab. 5 Results of different methods on random generated CVRP instances

方法	类型	VRP20			VRP50			VRP100		
		距离	Gap 值	总时间	距离	Gap 值	总时间	距离	Gap 值	总时间
Gurobi	Solver	6.10	0.00%			-			-	
LKH3	Solver	6.14	0.58%	7.2 ms	10.38	0.00%	25.2 ms	15.65	0.00%	46.8 ms
PtrNet [16]	RL, G	6.59	8.03%		11.39	9.78%		17.23	10.12%	
AM [8]	RL, G	6.40	4.97%	0.1 ms	10.98	5.86%	0.3 ms	16.80	7.34%	0.8 ms
Greedy	RL, G	6.26	2.6%	0.2 ms	10.88	4.81%	0.7 ms	16.69	6.68%	1.7ms
OR Tools	H, S	6.43	5.41%		11.31	9.01%		17.16	9.67%	
PtrNet [16]	SL, BS	6.40	4.92%		11.15	7.46%		16.96	8.39%	
AM [8]	RL, S	6.25	2.49%	36 ms	10.62	2.40%	0.17 s	16.23	3.72%	0.72 s
Sampling	RL, S	6.19	1.47%	84 ms	10.54	1.54%	0.44 s	16.16	3.25%	1.55 s

注：在类型列中，RL：强化学习方法、H:启发式方法、SL：监督学习、S：采样/搜索、G：贪婪搜索、BS：束波搜索、“-”：不能在合理时间内求解。

5.2.4 框架计算时间复杂度分析

该框架通过线下训练和线上测试的方式来求解路径问题。接下来，通过对 TSP 问题的求解(大多数文献[7][13]通过求解 TSP 来分析时间复杂度，为了便于与文献中的方法进行比较，本文也通过 TSP 进行评估)来评估所提出模型在训练和测试阶段随图规模(即节点数)增大和运行时间的关系。本节采用的所有算例均由均匀分布随机生成(与文献[7][13]保持一致)。对于训练阶段，训练时间不仅取决于问题的图规模，还取决于训练数据的数量和批量大小。为不失一般性，这里用 10 000 个训练实例和相同的批量大小(批量大小为 128)测试了单个 epoch 内节点数从 1 增加到 100 实例的运行时间。图 6 (a)显示了所提出的框架在训练阶段的运行时间随图节点增长呈线性增长。

在测试阶段，测试了图规模(节点数)从 1 增加到 500 时，整个编码器-解码器器模型的运行时间。图 6 (b)显示了所提出的模型在测试阶段的运行时间随图规模(节点数)的增加呈线性增长。表 6 总结了几类方法包括精确算法、启发式算法和基于学习的方法在规模为 100 个节点的 TSP 上的运行时间复杂度、运行时间(平均值)和平均最优解间距。除了所提出的框架的结果以外，所有的结果都取自 Drori 等[13]的表 1。所提出框架的结果以粗体列出。精确算法、近似算法、启发式算法的运行时间在图规模上至少呈平方增长，S2V-DQN[7]是强化学习方法，运行时间复杂度分别为 $O(n^2)$，且具有较大的最优解间距(为 8.4%)，本文所提出的框架的运行时间复杂度为 $O(n)$，在运行时间和最优解间距上都有较大的提升。GAT[13]与所提出的框架都具有相同的运行时间复杂度但其最优解间距更大。

表 6 各方法的运行时间复杂度

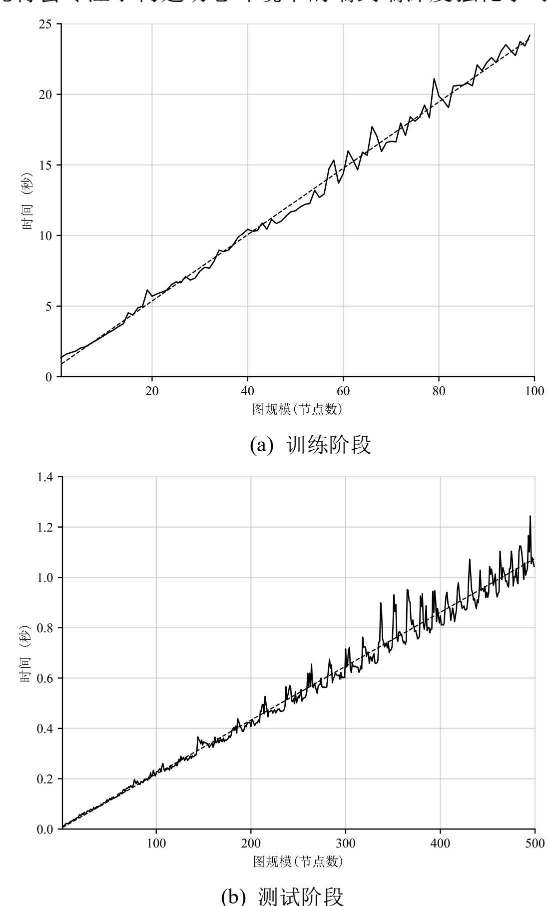
Tab. 6 Running time complexity of each method

方法	运行时间复杂度	运行时间 (毫秒)	Gap 值
Gurobi	NA	3, 220 ms	0.0%
Concorde	NA	254.1 ms	0.0%
Christofifides	$O(n^3)$	5, 002 ms	2.9%
LKH	$O(n^{2.2})$	2, 879 ms	0.0%
2-opt	$O(n^2)$	30.08 ms	9.7%
Farthest	$O(n^2)$	8.35 ms	7.5%
Nearest	$O(n^2)$	9.35 ms	24.5%
S2V-DQN[7]	$O(n^2)$	61.72 ms	8.4%
GAT[13]	$O(n)$	1.17 ms	7.4%
Greedy	$\mathbf{O(n)}$	**1.06** ms	**3.7%**

6 结束语

本文提出了端到端的深度强化学习框架用于提升 MDVRP 的求解效率。为 MDVRP 建模了马尔可夫决策过程，设计了改进图注意力网络作为编码器对求解过程中的状态信息进行编码，还基于 Transformer 模型设计了解码器模型。为训练所提出的编码器-解码器模型，设计了改进 REINFORCE 算法用于。此外，所设计的框架不受问题规模的约束，一旦模型完成训练就可以应用于不同规模的算例问题(不同的客户数和车场数)。为验证所提出框架的可行性和有效性，通过随机生成的算例和公开标准算例进行了数值实验，并与现有的 Learning-based 方法、Google OR-tools、元启发式算法进行对比。计算结果表明所提出的框架对于求解不同规模及不同场景下的车辆路径问题具有可行性和高效性。

本文考虑了静态环境下的 MDVRP，而在实际的物流运输过程中，运输环境通常是瞬息万变地，即会面临订单动态到达的情况。基于本文所提出框架求解该问题的快速性，其具有在动态环境下进行实时调度车辆的潜力。因此，未来的研究将会专注于构建动态环境下的端到端深度强化学习框架。



(a) 训练阶段

(b) 测试阶段

图 6 所提出框架的运行时间复杂度分析

Fig. 6 Running time complexity analysis of the proposed framework

参考文献：

[1] Sharma N, Monika. A Literature Survey on Multi Depot Vehicle Routing Problem [J]. International Journal for Scientific Research Development, 2015, 3 (4): 1752-1757.

[2] 叶勇，张惠珍. 多配送中心车辆路径问题的狼群算法 [J]. 计算机应用研究, 2017, 34 (9): 2590-2593. (Ye Yong, Zhang Huizhen. Wolf pack algorithm for multi-depot vehicle routing problem [J]. Application Research of Computers, 2017, 34 (9): 2590-2593.)

[3] 胡蓉，陈文博，钱斌等. 学习型蚁群算法求解绿色多车场车辆路径问题 [J]. 系统仿真学报, 2021, 33 (9): 2095-2108. (Hu Rong, Chen Wenbo, Qian Bin, *et al.* learning ant colony algorithm for green multi-depot vehicle routing problem [J]. Journal of System Simulation, 2021, 33 (9): 2095-2108.)

[4] 戚远航，蔡延光，蔡颢，等. 泰森多边形的离散蝙蝠算法求解多车场车辆路径问题 [J]. 控制理论与应用, 2018, 35 (8): 1142-1150. (Qi Yuanhang, Cai Yanguang, Cai Hao, *et al.* Voronoi diagram-based discrete bat algorithm formulti-depot vehicle routing problem [J]. Control Theory & Applications, 2018, 35 (8): 1142-1150.)

[5] 李洋，胡蓉，钱斌等. 两阶段算法求解多车场车辆路径问题 [J]. 信息与控制, 2020, 49 (6): 752-760. (Li Yang, Hu Ron, Qian Bin, *et al.* two-stage algorithm for multi-depot vehicle routing problem [J]. Information and Control, 2020, 49 (6): 752-760.)

[6] Bengio Y, Lodi A, Prouvost A. Machine learning for combinatorial optimization: A methodological tour d'horizon [J]. European Journal of Operational Research, 2021, 290 (2): 405-421.

[7] Khalil E, Dai H, Zhang Y, *et al.* Learning combinatorial optimization algorithms over graphs [J]. In Advances in Neural Information Processing Systems, 2017, 30: 6348-6358.

[8] Kool W, Van H H, Welling M. Attention, Learn to Solve Routing Problems! [C]// proceedings of the International Conference on Learning Representations, 2018.

[9] Malazgirt G A, Unsal O S, Kestelman A C. Tauriel: Targeting traveling salesman problem with a deep reinforcement learning inspired architecture [J/OL]. 2019, arXiv preprint arXiv: 1905. 05567.

[10] Ma Q, Ge S, He D, *et al.* Combinatorial Optimization by Graph Pointer Networks and Hierarchical Reinforcement Learning [J/OL]. 2019, arXiv: 1911. 04936.

[11] Cappart Q, Moisan T, Rousseau L M, *et al.* Combining Reinforcement Learning and Constraint Programming for Combinatorial Optimization [J/OL]. 2020, arXiv: 2006. 01610.

[12] Subramaniam V, Lee G K, Ramesh T, *et al.* Machine Selection Rules in a Dynamic Job Shop [J]. The International Journal of Advanced Manufacturing Technology, 2000, 16 (12): 902-908.

[13] Drori I, Kharkar A, Sickinger W R, *et al.* Learning to Solve Combinatorial Optimization Problems on Real-World Graphs in Linear Time [J/OL]. 2020, arXiv: 2006. 03750.

[14] Zhang R, Prokhorchuk A, Dauwels J. Deep Reinforcement Learning for Traveling Salesman Problem with Time Windows and Rejections [C]// International Joint Conference on Neural Networks (IJCNN) , 2020: 1-8.

[15] Hu Y, Yao Y, Lee W S. A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs [J]. Knowledge-Based Systems, 2020, 204: 106244.

[16] Nazari M, Oroojlooy A, Snyder L, *et al.* Reinforcement learning for solving the vehicle routing problem [C]// Advances in Neural Information Processing Systems, 2018: 9839-9849.

[17] Chen X, Tian Y. Learning to perform local rewriting for combinatorial optimization [C]// proceedings of the Advances in Neural Information Processing Systems, 2019: 6281-6292.

[18] Zhao J, Mao M, Zhao X, *et al.* A Hybrid of Deep Reinforcement Learning and Local Search for the Vehicle Routing Problems [J]. Ieee Transactions on Intelligent Transportation Systems, 2020: 1-11.

[19] Lu H, Zhang X, Yang S. A Learning-based Iterative Method for Solving Vehicle Routing Problems [C]// proceedings of the International Conference on Learning Representations, 2019.

[20] Gao L, Chen M, Chen Q, *et al.* Learn to Design the Heuristics for Vehicle Routing Problem [J/OL]. 2020, arXiv: 2002. 08539.

[21] Chen M, Gao L, Chen Q, *et al.* Dynamic Partial Removal: A Neural Network Heuristic for Large Neighborhood Search [J/OL]. 2020, arXiv: 2005. 09330.

[22] Zhang K, He F, Zhang Z, *et al.* Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach [J]. Transportation Research Part C: Emerging Technologies, 2020, 121: 102861.

[23] 王万良，陈浩立，李国庆，等. 基于深度强化学习的多配送中心车辆路径规划 [J/OL]. 控制与决策: 1-9 [2022-04-21]. DOI: 10. 13195/j. kzyjc. 2021. 1381. (Wang Wanliang, Chen Haoli, Li Guoqing, *et al.* Deep Reinforcement Learning for Multi-depot Vehicle Routing Problem [J/OL]. Control and Decision: DOI: 10. 13195/j. kzyjc. 2021. 1381.)

[24] Vinyals O, Fortunato M, Jaitly N. Pointer networks [C]// proceedings of the Advances in Neural Information Processing Systems, 2015: 2692-2700.

[25] Bello I, Pham H, Le Q V, *et al.* Neural combinatorial optimization with reinforcement learning [J/OL]. arXiv preprint arXiv: 09940, 2016.

[26] Li Z, Chen Q, Koltun V. Combinatorial optimization with graph convolutional networks and guided tree search [C]// proceedings of the Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018: 537-546.

[27] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [J/OL]. arXiv preprint arXiv: 02907, 2016.

[28] Mnih V, Kavukcuoglu K, Silver D, *et al.* Playing Atari with Deep Reinforcement Learning [J/OL]. 2013, arXiv: 1312. 5602.

[29] Dai H, Dai B, Song L. Discriminative embeddings of latent variable models for structured data [J]. International conference on machine learning, 2016: 2702-2711.

[30] Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need [C]// In Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.

[31] Nowak A, Villar S, Bandeira A S, *et al.* A note on learning algorithms for quadratic assignment with graph neural networks [C]// proceedings of the Proceeding of the 34th International Conference on Machine Learning (ICML) , 2017: 22.

[32] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]// Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37. Lille, France; JMLR. org. 2015: 448–456

[33] Velickovic P, Cucurull G, Casanova A, *et al.* Graph Attention Networks [C]// proceedings of the International Conference on Learning Representations, 2018.

[34] Van H H, Guez A, Silver D. Deep Reinforcement Learning with Double Q-Learning [C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2016, 30 (1) .

[35] Cordeau J F, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems [J]. Networks, 1997, 30 (2): 105-119.

[36] Cordeau J F, Maischberger M. A parallel iterated tabu search heuristic for vehicle routing problems [J]. Computers & Operations Research, 2012, 39 (9): 2033-2050.

[37] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization [C]// In International Conference on Learning Representations, 2015.

[38] Stodola P. Using Metaheuristics on the Multi-Depot Vehicle Routing Problem with Modified Optimization Criterion [J]. 2018, 11 (5): 74.